

Programmieren von Klassen in Java

- Installation von Java
- Einführung in die Programmiersprache Java
- Grundelemente einer Klasse in Java
- Attribute in Java
- Methoden in Java
- main-Methode: Startpunkt eines Java-Programms

02.11.2016

Monika Tepfenhart

1

Installation von Java

- Erweitern der Umgebungsvariable path:
<https://docs.oracle.com/javase/tutorial/essential/environment/paths.html>

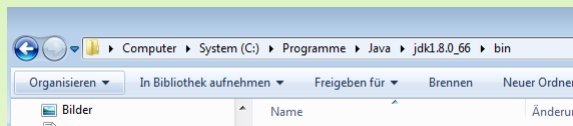
02.11.2016

Monika Tepfenhart

2

Installation von Java – Windows 7

- Java Installationspfad kopieren



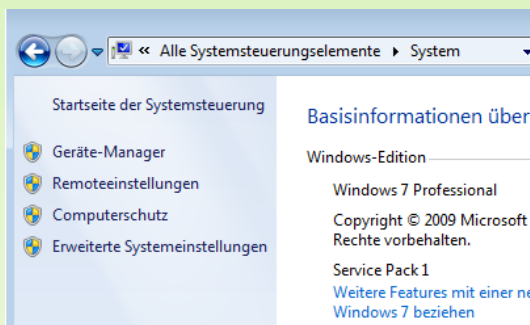
02.11.2016

Monika Tepfenhart

3

Installation von Java – Windows 7

- Start – Systemsteuerung - System



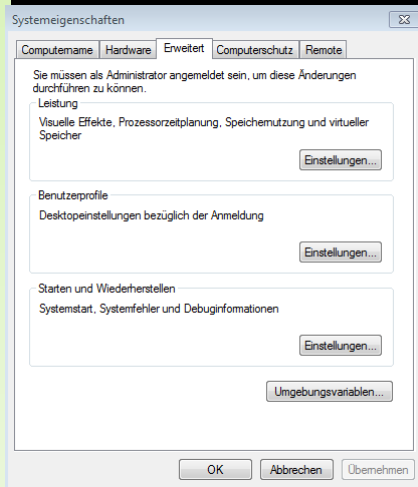
- Auswahl: Erweiterte Systemeinstellungen

02.11.2016

Monika Tepfenhart

4

Installation von Java – Windows 7



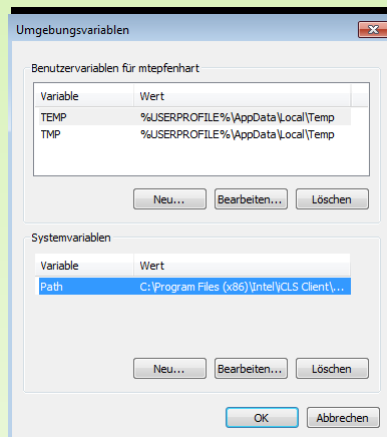
- Auswahl:
Umgebungsvariablen

02.11.2016

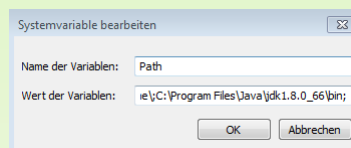
Monika Tepfenhart

5

Installation von Java – Windows 7



- Variable Path
Bearbeiten und erweitern:
„;“ + **Java Installationspfad** + „;“



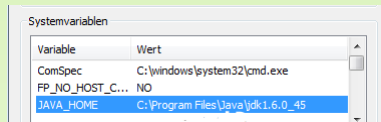
02.11.2016

Monika Tepfenhart

6

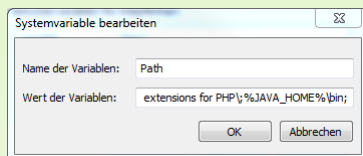
Installation von Java – Windows 7

- Umgebungsvariable **JAVA_HOME** setzen



- Variable Path Bearbeiten und erweitern:

„;“ + „%“ + **JAVA_HOME** + „%“ + „\bin“ + „;“



02.11.2016

Monika Tepfenhart

7

Installation von Java – openSUSE 42.1

- Installation von Java in das **/usr** Verzeichnis

```
linux-insc:/usr # ls -ar --format=long
total 4
drwxr-xr-x 1 root root    26 Jul 11 10:42 x86_64-suse-linux
lrwxrwxrwx 1 root root    10 Jul 11 10:42 tmp -> ../var/tmp
drwxr-xr-x 1 root root   112 Jul 12 14:23 src
drwxr-xr-x 1 root root  4134 Jul 11 15:09 share
drwxr-xr-x 1 root root 11386 Jul 12 14:18 sbin
drwxr-xr-x 1 root root    76 Jul 11 10:42 local
drwxr-xr-x 1 root root 115630 Jul 12 14:19 lib64
drwxr-xr-x 1 root root   5906 Jul 11 15:07 lib
drwxr-xr-x 1 root root    66 Jul 11 11:28 java
drwxr-xr-x 1 root root   2594 Jul 12 14:19 include
drwxr-xr-x 1 root root     0 Sep 30 2015 games
drwxr-xr-x 1 root root   100 Jul 11 11:48 eclipse
drwxr-xr-x 1 root root  50708 Jul 12 14:19 bin
drwxr-xr-x 1 root root    10 Jul 11 10:42 X11R6
drwxr-xr-x 1 root root   166 Jul 11 10:53 ..
drwxr-xr-x 1 root root   152 Jul 11 11:15 .
```

02.11.2016

Monika Tepfenhart

8

Installation von Java – openSUSE 42.1

➤ Installation von Java in das `/usr` Verzeichnis

```
linux-insc:/usr # ls -ar --format=long java/
total 0
drwxr-xr-x 1 uucp 143 300 Jul 11 11:30 jdk1.8.0_92
drwxr-xr-x 1 uucp 143 278 Jul 11 11:27 jdk1.7.0_79
drwxr-xr-x 1 root root 164 Mar 27 2013 jdk1.6.0_45
drwxr-xr-x 1 root root 152 Jul 11 11:15 ..
drwxr-xr-x 1 root root 66 Jul 11 11:28 .
```

02.11.2016

Monika Tepfenhart

9

Installation von Java – openSUSE 42.1

➤ Installation von Java in das `/usr` Verzeichnis

```
linux-insc:/usr # cd java/jdk1.6.0_45/
linux-insc:/usr/java/jdk1.6.0_45 # ls -ar --format=long
total 18676
-rw-r--r-- 1 root root 18936852 Mar 26 2013 src.zip
drwxr-xr-x 1 root root 34 Mar 27 2013 man
drwxr-xr-x 1 root root 172 Jul 11 11:21 lib
drwxr-xr-x 1 root root 182 Jul 11 11:21 jre
drwxr-xr-x 1 root root 118 Mar 27 2013 include
drwxr-xr-x 1 root root 166 Mar 27 2013 db
drwxr-xr-x 1 root root 594 Mar 27 2013 bin
-r--r--r-- 1 root root 168902 Mar 26 2013 THIRDPARTYLICENSEREADME.txt
-r--r--r-- 1 root root 115 Mar 26 2013 README.html
-r--r--r-- 1 root root 40 Mar 26 2013 LICENSE
-r--r--r-- 1 root root 3339 Mar 26 2013 COPYRIGHT
drwxr-xr-x 1 root root 66 Jul 11 11:28 ..
drwxr-xr-x 1 root root 164 Mar 27 2013 .
```

02.11.2016

Monika Tepfenhart

10

Installation von Java – openSUSE 42.1

- Anpassen der .bashrc Datei im eigenen Home Verzeichnis

```
mtepfenh@linux-insc:~> pwd
/home/mtepfenh
mtepfenh@linux-insc:~> ls -al
insgesamt 152
drwxr-xr-x 27 mtepfenh users 4096 29. Jul 09:02 .
drwxr-xr-x  3 root    root    22 11. Jul 11:03 ..
-rw-----  1 mtepfenh users 15560 26. Jul 15:41 .bash_history
-rw-r--r--  1 mtepfenh users 1383 11. Jul 12:59 .bashrc
```

- Setzen der JAVA_HOME Variable & Erweitern der PATH Variable

```
export JAVA_HOME=/usr/java/jdk1.6.0_45
#export JAVA_HOME=/usr/java/jdk1.7.0_79
#export JAVA_HOME=/usr/java/jdk1.8.0_92
export PATH=$JAVA_HOME/bin:$PATH
```

02.11.2016

Monika Tepfenhart

11

Einführung in die Programmiersprache Java

- Java ist plattform-unabhängig
 - Ausführbar auf Linux-, Windows- oder Mac-Rechner
 - Anpassungen ans Betriebssystem nicht notwendig
 - „Java-Laufzeitumgebung“ (auch: Java Runtime Environment, kurz: JRE) – ermöglicht die Ausführung auf verschiedenen Plattformen
- Java-Compiler übersetzt Quelltext „Bytecode“
 - „Bytecode“ wird vom JRE geladen, gestartet und ausgeführt

02.11.2016

Monika Tepfenhart

12

Einführung in die Programmiersprache Java

- **Java-Runtime** besteht aus:
 - Java Virtual Machine (kurz: JVM)
 - Java Klassenbibliothek
- **Java Virtual Machine (JVM)** führt Bytecode aus
- **Java Software Development Kit (SDK)** ist das Grundwerkzeug des Java-Entwicklers. Enthält:
 - Compiler und JVM

02.11.2016

Monika Tepfenhart

13

Einführung in die Programmiersprache Java

- Java Klassen werden Textdateien *.java gespeichert
 - Jede Klasse wird eine eigene Datei angelegt
 - Klassenname und Dateiname sind identisch
- Java-Compiler erzeugt aus jeder Klasse eine Klasse (kompilierter Bytecode) mit der Endung *.class
- Kompilieren von HelloWorld.java: `javac HelloWorld.java`
- Ausführen von HelloWorld.class: `java HelloWorld`

02.11.2016

Monika Tepfenhart

14

Einführung in die Programmiersprache Java

➤ Die Klasse „HelloWorld.java“

```
1 public class HelloWorld {  
2  
3     public static void main(String[] args) {  
4         System.out.println("Hallo");  
5     }  
6 }
```

02.11.2016

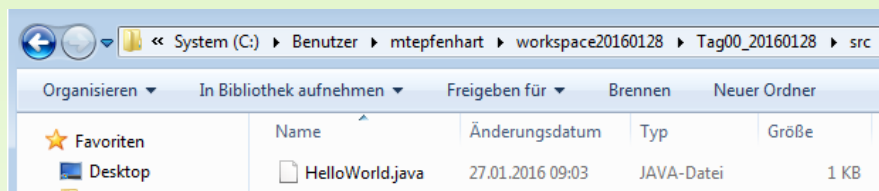
Monika Tepfenhart

15

Einführung in die Programmiersprache Java

➤ Die Klasse „HelloWorld.java“ im Verzeichnis

```
C:\Users\mtepfenart\workspace20160128\Tag00_20160128\src>dir  
Datenträger in Laufwerk C: ist System  
Volumeseriennummer: CA1C-6790  
  
Verzeichnis von C:\Users\mtepfenart\workspace20160128\Tag00_20160128\src  
27.01.2016  09:08    <DIR>          .  
27.01.2016  09:08    <DIR>          ..  
27.01.2016  09:03             115 HelloWorld.java  
               1 Datei(en),             115 Bytes  
               2 Verzeichnis(se), 404.718.698.496 Bytes frei
```



02.11.2016

Monika Tepfenhart

16

Einführung in die Programmiersprache Java

➤ Kompilieren von „*HelloWorld.java*“

```
C:\Users\mtepfenhard\workspace20160128\Tag00_20160128\src>javac HelloWorld.java
C:\Users\mtepfenhard\workspace20160128\Tag00_20160128\src>
```

➤ Kompilierter ByteCode „*HelloWorld.class*“

```
C:\Users\mtepfenhard\workspace20160128\Tag00_20160128\src>dir
Datenträger in Laufwerk C: ist System
Volumeserienummer: CA1C-6790

Verzeichnis von C:\Users\mtepfenhard\workspace20160128\Tag00_20160128\src

27.01.2016  09:13    <DIR>          .
27.01.2016  09:13    <DIR>          ..
27.01.2016  09:16                419 HelloWorld.class
27.01.2016  09:03                115 HelloWorld.java
                2 Datei(en),          534 Bytes
                2 Verzeichnis(se), 404.712.542.208 Bytes frei
```

02.11.2016

Monika Tepfenhart

17

Einführung in die Programmiersprache Java

➤ Kompilierter ByteCode „*HelloWorld.class*“

« System (C:) » Benutzer » mtepfenhard » workspace20160128 » Tag00_20160128 » src				
Organisieren ▼	In Bibliothek aufnehmen ▼	Freigeben für ▼	Brennen	Neuer Ordner
Favoriten				
Desktop				
Downloads				
Name	Änderungsdatum	Typ	Größe	
HelloWorld.class	27.01.2016 09:16	CLASS-Datei	1 KB	
HelloWorld.java	27.01.2016 09:03	JAVA-Datei	1 KB	

02.11.2016

Monika Tepfenhart

18

Einführung in die Programmiersprache Java

- Ausführen des kompilierten ByteCodes
„*HelloWorld.class*“

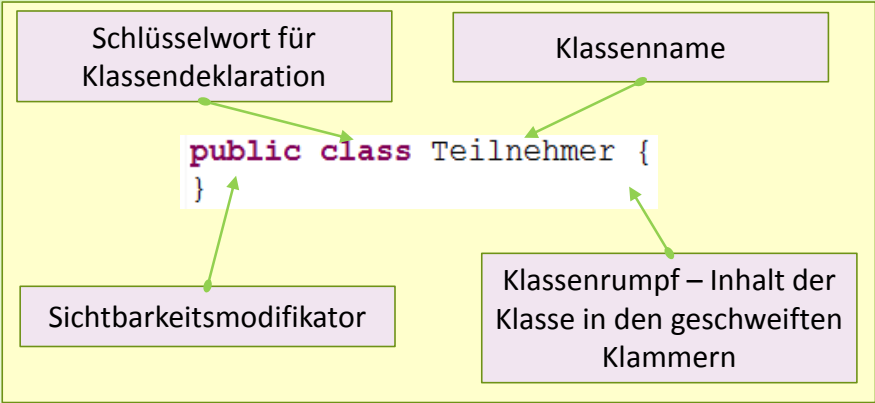
```
C:\Users\mtepfenart\workspace20160128\Tag00_20160128\src>java HelloWorld  
Hallo
```

Grundelemente einer Klasse in Java

- Eine Klasse liefert Struktur „Vorlage“ zur Bildung eines „Objektes“
- Aus einer Klasse erzeugten Objekte haben somit die gleiche Struktur
 - die gleichen Attribute
 - verfügen über die gleichen Methoden (Funktionen)
 - Alle Objekte einer Klasse können somit in gleicher Weise verarbeitet werden

Grundelemente einer Klasse in Java

Gerüst einer Klasse in Java:



Grundelemente einer Klasse in Java

Element einer Klasse	Beschreibung	Beispiel
Sichtbarkeitsmodifikator	Legt Sichtbarkeit der Klasse für andere Klassen fest	public
Schlüsselwort für die Klassendeklaration	Zeigt dem Java-Compiler an, dass im Folgenden eine Java-Klasse programmiert ist	class
Klassenname	Legt Namen für die Klasse in Java fest und wird als Dateiname verwendet.	Kunde
Geschweifte Klammern	Markieren Inhalt einer Klasse	{ ... }

Attribute in Java

- **Attribute (Datenfelder oder Instanzvariablen)**
 - (auch: Eigenschaft von Klassen, engl. property) sind statische Elemente von Klassen
- In einem Attribut können konkrete Werte gespeichert werden.
- Attribute dienen ausschließlich zum Speichern von Werten.

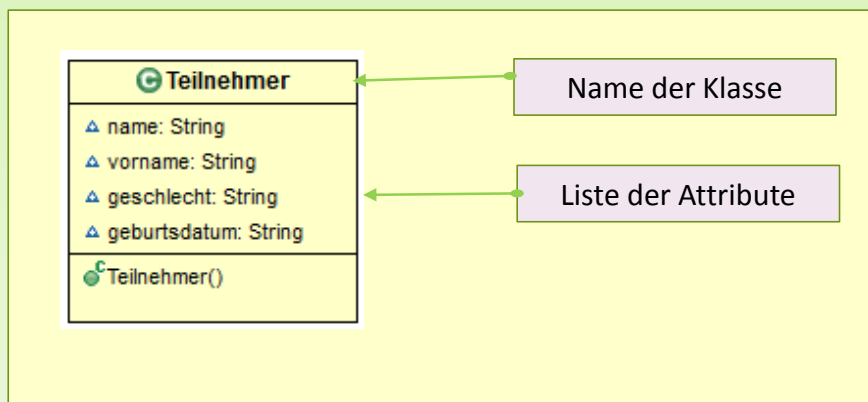
02.11.2016

Monika Tepfenhart

23

Attribute in Java

UML Klassendiagramm der Klasse „Teilnehmer“



02.11.2016

Monika Tepfenhart

24

Attribute in Java

Quelltext der Klasse „Teilnehmer“

```
public class Teilnehmer {  
    String name;  
    String vorname;  
    String geschlecht;  
    String geburtsdatum;  
}
```

Attribute in Java

Eigenschaften des Attributs	Beschreibung	Beispiel
Name	Name des Attributs	Vorname
Datentyp	Legt fest, wie die Werte zu dem Attribut aussehen, also ob es z. B. eine Zahl, eine Zeichenkette oder ein Datum ist	Date (Datum) String (Zeichenkette) Integer (Ganze Zahl)
Konstante (ja/nein)	Legt fest, ob sich der Wert des Attributs ändern darf oder nicht	Ger. Kreiszahl Pi: 3,1415
Defaultwert	Voreingestellter Wert	01.01.2010

Methoden in Java

- **Methoden** (auch: **Funktionen**, **Operationen**)
 - dynamische Elemente von Klassen
 - Sie enthalten Algorithmen, Anweisungen und Abarbeitungsvorschriften, mit denen Werte erstellt, berechnet, verändert und gelöscht werden können
 - Die Ergebnisse der Methoden können in Attribute der Klasse gespeichert werden
-

02.11.2016

Monika Tepfenhart

27

Methoden in Java

- Methoden bestehen aus **Kopf** und **Rumpf**
 - Rumpf folgt dem Kopf
 - Rumpf ist in geschweiften Klammern
 - Rumpf enthält:
 - Deklarationen (temp. Speicher für Variablen)
 - Anweisungen (beschreiben Aktionen der Methoden)
 - Alle Deklarationen und Anweisungen zwischen den geschweiften Klammern werden als **Block** bezeichnet
-

02.11.2016

Monika Tepfenhart

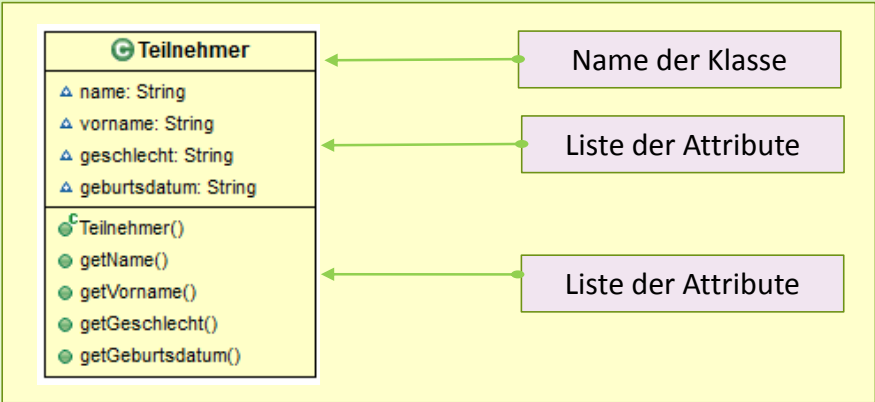
28

Methoden in Java

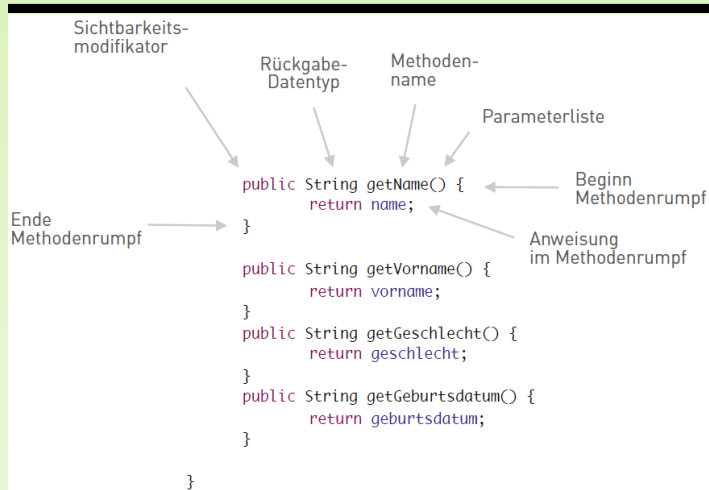
Elemente von Methoden	Beschreibung	Beispiel
Name	Methodenname	getName()
Parameter	Benötigte Objekte und Werte, die zur Abarbeitung der Methode erforderlich sind	(String name, String vorname) (Integer zahl1, Integer zahl2)
Rückgabewert	Angabe des Datentyps des Objektes, in dem das Ergebnis der Methode gespeichert wird	Date (Datum) String (Zeichenkette) Integer (Ganze Zahl)

Methoden in Java

UML Klassendiagramm der Klasse „Teilnehmer“



Methoden in Java



The diagram shows a Java class with four methods. Arrows point from labels to specific parts of the code:

- Sichtbarkeits-modifikator** points to `public` in the first method.
- Rückgabe-Datentyp** points to `String` in the first method.
- Methoden-name** points to `getName()` in the first method.
- Parameterliste** points to the empty parentheses in the first method.
- Beginn Methodenrumpf** points to the opening curly brace of the first method.
- Anweisung im Methodenrumpf** points to the `return name;` statement in the first method.
- Ende Methodenrumpf** points to the closing curly brace of the first method.

```
public String getName() {  
    return name;  
}  
  
public String getVorname() {  
    return vorname;  
}  
  
public String getGeschlecht() {  
    return geschlecht;  
}  
  
public String getGeburtsdatum() {  
    return geburtsdatum;  
}  
}
```

02.11.2016

Monika Tepfenhart

31

Methoden in Java

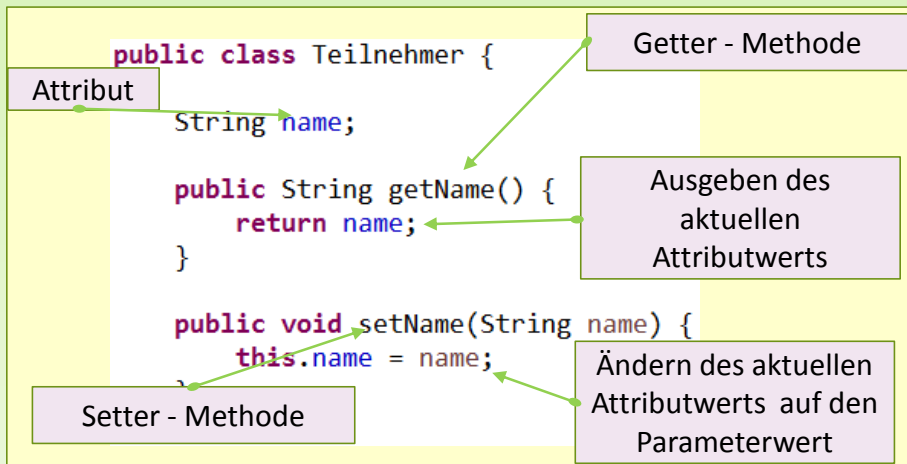
- **Methodennamen**
 - können beliebig sein
 - an den Namen sollte die Funktion erkennbar sein
- **Getter- und Setter- Methoden**
 - Schreiben und Auslesen von Attributen geschieht mit
- **Getter-Methode („get“ = „holen“)**
 - liefert den Wert eines Attributs zurück
- **Setter-Methode („set“ = „setzen“)**
 - ändert den Wert eines Attributs auf den Wert, der als Parameter der Setter-Methode übergeben wird

02.11.2016

Monika Tepfenhart

32

Methoden in Java



02.11.2016

Monika Tepfenhart

33

Methoden in Java

- **Sondierende Methoden** liefern Informationen über den Zustand des gerufenen Objekts
 - Haben Rückgabeeinweisung, ein Wert wird zurückgeliefert
 - Zurückgeben bedeutet: Eine Information wird zwischen zwei Programmteilen weitergereicht
 - Ergebnistyp und Rückgabeeinweisung bedingen einander

```
public String getName() {  
    return name;  
}
```

02.11.2016

Monika Tepfenhart

34

Methoden in Java

- **Verändernde Methoden** ändern beim Aufruf den Wert eines oder mehrerer Datenfelder (Attribute)
 - Methode übernimmt Parameter, Wert des Parameters überschreibt Wert der Datenfelder (Attribute)
 - Verhalten des Objekts (Klasse) ändert sich durch den Aufruf
 - Haben den Ergebnistyp void – Methode liefert keinen Wert an ihren Aufrufer zurück

```
public void setName(String name) {  
    this.name = name;  
}
```

02.11.2016

Monika Tepfenhart

35

Methoden in Java

- **Methode mit Parametern:**
 - übernimmt Daten
 - verwendet Daten um bestimmte Aufgabe auszuführen
- **Methode ohne Parametern:**
 - Verwendet Daten aus Datenfeldern (Attributen)

02.11.2016

Monika Tepfenhart

36

Methoden in Java

➤ Methode mit Ergebnistyp:

- liefert den Wert dorthin zurück, von wo sie gerufen wird
- verwendet Daten um bestimmte Aufgabe auszuführen

➤ Methode ohne Ergebnistyp (void):

- liefern Nichts zurück

02.11.2016

Monika Tepfenhart

37

main-Methode eines Java-Programms

-
- Für jedes Java-Programm gibt es einen Startpunkt
 - Der Startpunkt ist eine Methode, sie wird immer als erstes aufgerufen, wenn das Programm startet:
 - die main-Methode

```
public static void main(String[] args) {  
}
```

02.11.2016

Monika Tepfenhart

38

main-Methode eines Java-Programms

Deklaration der main -
Methode

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hallo");  
    }  
}
```

Ende der main -
Methode

Beginn der main -
Methode

02.11.2016

Monika Tepfenhart

39